CLAIMS

1. A method of developing an application utilizing a device-independent programming language, comprising:

determining a functionality of an current application, the current application being desired to be accessed via at least one communication device; and

defining a device-independent application representation of the functionality of the current application in the device-independent programming language;

wherein the device-independent programming language can be interpreted by a system to allow for the generation and transmission of a device-specific programming language for each of the at least one communication devices.

2. The method of Claim 1, wherein the step of defining the device-independent application representation further comprises:

defining at least one process, each of the at least one processes detailing a task to be completed in accordance with the device-independent application representation;

for each of the defined processes, defining at least one process step, each of the at least one process steps combining to realize the defined process;

for each of the defined process steps, defining at least one user interaction, each of the at least one user interactions combining to realize the defined process step; and

for each of the defined user interactions, defining at least one element, each of the at least one elements combining to realize the defined interactions.

3. The method of Claim 2, further comprising storing the device-independent application representation and allowing the memory location to be accessed.

4. The method of Claim 1, wherein the current application corresponds to a set of functionality to interact with an electronic application and one of the at least one communication devices.

5. The method of Claim 1, wherein each of the at least one communication devices includes a device presentation language.

6. The method of Claim 1, further comprising performing an invocation call to a back end system to access the functionality of the back end application.

7. The method of Claim 6, further comprising transferring data contained within the device-independent application representation, received from the communication device, to the back end system.

8. The method of Claim 6, further comprising receiving data from the back end system containing the device-independent application representation.

9. The method of Claim 1, wherein the device-independent programming language is an extensible markup language (XML).

10. The method of Claim 1, wherein the device-independent programming language is a multiple-channel access extensible markup language (MAXML™).

11. The method of Claim 1, wherein the device-independent application representation corresponds to a human-information interaction model.

12. A system for developing an application utilizing a device-independent programming language, comprising:

means for determining a functionality of an current application, the current application being desired to be accessed via at least one communication device; and

means for defining a device-independent application representation of the functionality of the current application in the device-independent programming language;

wherein the device-independent programming language can be interpreted by a system to allow for the generation and transmission of a device-specific programming language for each of the at least one communication devices.

13. The system of Claim 12, wherein the means for defining the device-independent application representation further comprises:

means for defining at least one process, each of the at least one processes detailing a task to be completed in accordance with the device-independent application representation;

for each of the defined processes, means for defining at least one process step. each of the at least one process steps combining to realize the defined process;

for each of the defined process steps, means for defining at least one user interaction, each of the at least one user interactions combining to realize the defined process step; and

for each of the defined user interactions, means for defining at least one element, each of the at least one elements combining to realize the defined interactions.

14. The system of Claim 13, further comprising means for storing the device-independent application representation and means for allowing the memory location to be accessed.

15. The system of Claim 12, further comprising means for performing an invocation call to a back end system to access the functionality of the back end application.

16. The system of Claim 17, further comprising means for transferring data contained within the device-independent application representation, received from the communication device to the back end system.

17. The system of Claim 17, further comprising means for receiving data from the back end system containing the device-independent application representation.

18. The method of Claim 1, wherein the device-independent application representation corresponds to a human-information interaction model.

19. A device-independent programming language, comprising:

a device-independent application representation, the device-independent application representation being related to a task to be implemented between an electronic application and at least one communication device;

wherein the device-independent programming language can be interpreted by a system to allow for the generation and transmission of a device-specific programming language for each of the at least one communication devices.

20. The method of Claim 1, wherein the device-independent application representation corresponds to a human-information interaction model.